Looking to Relations for Future Trajectory Forecast - Supplementary Material

Chiho Choi
Honda Research Institute USA
cchoi@honda-ri.com

Behzad Dariush
Honda Research Institute USA
bdariush@honda-ri.com

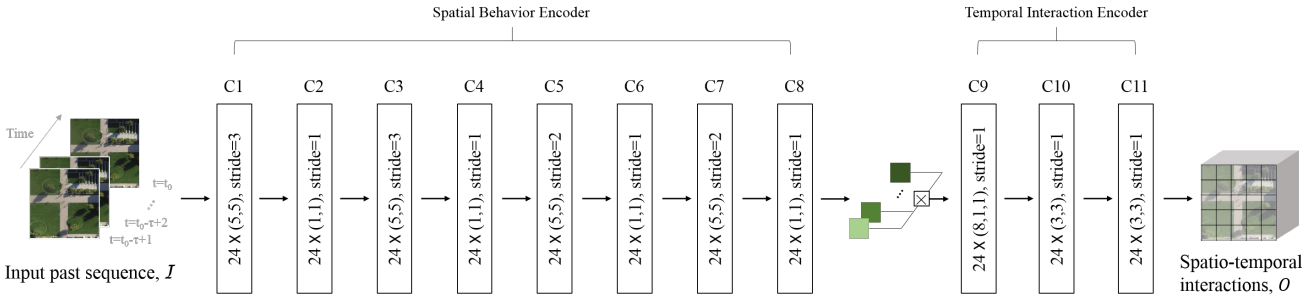


Figure 9: The detailed structure of the proposed spatial behavior encoder and temporal interaction encoder of our gated relation encoder. Given τ number of past images, the network first encodes spatial features using a set of 2D convolutional layers (C1 − C8) from each region of the discretized grid. The resulting features $\{s_1, ..., s_n\}$ are concatenated along the time axis to form S. Then, we run a 3D convolution (C9) with additional 2D convolutions (C10 − C11) to extract spatio-temporal interactions O. The inscribed numbers denote [the number of filters x (its width, height), and stride]. For 3D convolution, we use (its depth, width, height). Every layer comes with a ReLU activation function at the end. We also use a dropout layer after C6 (with a ratio 0.2) and C8 (with a ratio 0.5).

This document serves as supplementary materials to our paper *Looking to Relations for Future Trajectory Forecast*.

7. Architecture

7.1. Gated Relation Encoder

We visualize the network architecture of the proposed gated relation encoder (GRE) in Figure 9 and Figure 10. The network first encodes frame-wise feature representations of the road structures, the road topology, and the appearance of road users through 2D convolutional layers (C1−C8) as shown in Figure 9. Then, we concatenate spatial features along the time axis to construct a spatial feature description S that knowledge road users and road structures in each region of discretized grid. The subsequent 3D convolutional layer models transition patterns of each spatial region and build spatio-temporal representations O. Each cell o_i of $O = \{o_1, ..., o_n\}$ contains an interpretation of spatial behavior of all road users in a specific grid space and their temporal interactions with the environment.

The subsequent relation gate module (RGM) displayed in Figure 10. The RGM is designed to infer pair-wise relations from spatio-temporal interactions (*i.e.*, objects). A

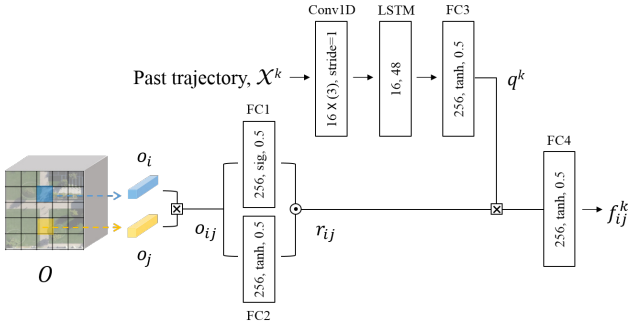


Figure 10: The structure of the proposed relation gate module. The notations are as follows: FC - fully connected [output size, activation function, dropout ratio], Conv1D - 1D convolution [the number of filters x (its width), stride], and LSTM - long short term memory unit [length, output size]. The symbols for operations are shown in the main manuscript.

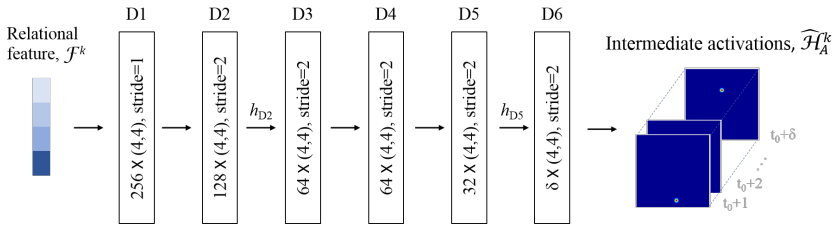


Figure 11: The architecture of the trajectory prediction network used to generate δ number of activations $\widehat{\mathcal{H}}_A^k$ which correspond to δ future locations. D is a deconvolutional layer and h is an output feature map. The inscribed numbers denote [the number of filters x (its width, height), stride]. Each deconvolutional layer comes with a ReLU activation function at the end.

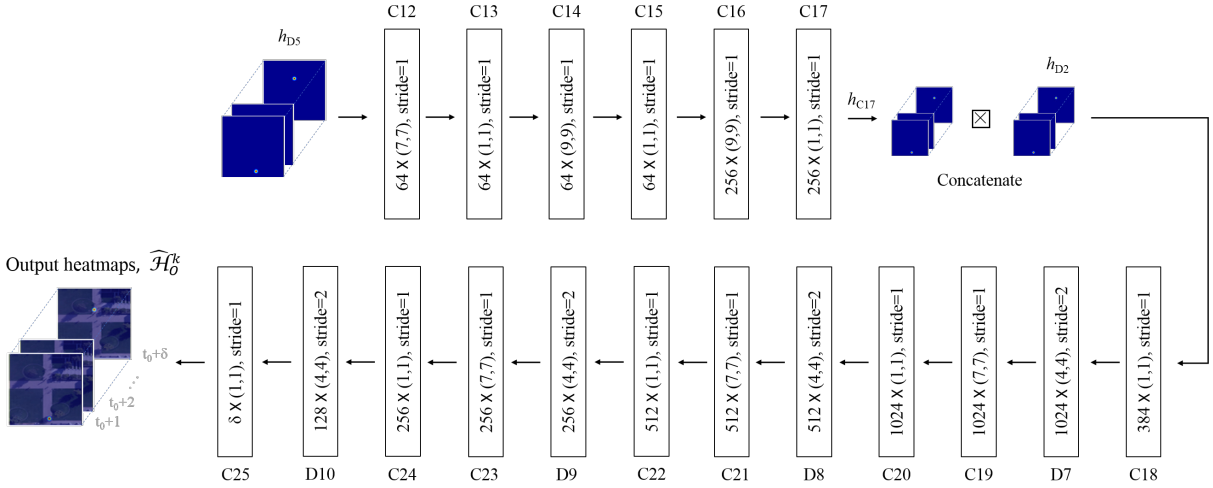


Figure 12: The architecture of the spatial refinement network. In the first stage, C12−C17 graudally decreases the dimensionality of h_{D5}. We use a max pooling layer after C13, C15, and C17. Next, the concatenated activations $h_{C17} \boxtimes h_{D2}$ are upsampled through four deconvolutional layers that come with a 1 x 1 and 7 x 7 convolution. We use ReLU activations with all convolutional and deconvolutional layers.

pair of objects (o_i, o_j) is inputted into our RGM to determine whether relations between the given objects would affect the target road users potential path through FC1 and FC2. By considering the motion history q^k of the target road user k, the RGM identifies how their relations can affect the future motion of the target k based on its past motion context q^k. We collect all relational information f_{ij}^k and perform element-wise sum to produce target-specific relational features $\mathcal{F}^k$.

7.2. Trajectory Prediction Network

Given a relational state $\mathcal{F}$, our trajectory prediction network (TPN) in Figure 11 constructs δ number of activations $\mathcal{A}$ which can be used to compute future locations. For this, we use six deconvolutional layers to upsample the dimensionality of the intermediate features. At the end of this process, we get $\mathcal{A} \in \mathbb{R}^{128 \times 128 \times \delta}$ and find all points with the maximum likelihood to locate future motion.

7.3. Spatial Refinement Network

The intermediate features h_{D2} and h_{D5} of the TPN are used to learn spatial dependencies between initial predictions $\widehat{\mathcal{H}}_A^k$. In the spatial refinement network (SRN), we first downsample $h_{D5} \in \mathbb{R}^{64 \times 64 \times 32}$ to be of size $h_{C17} \in \mathbb{R}^{8 \times 8 \times 256}$ using six convolutional layers (C12−C17, max pooling after C13, C15, and C17) as shown in Figure 12. Then, the concatenated features $h_{C17} \boxtimes h_{D2}$ are fed to deconvolutional layers (D7−D10) with 1 x 1 and 7 x 7 convolutions. In this process, we can successfully capture spatial dependencies between future locations by using large receptive fields, increasing the number of layers, and conducting pixel-level correction. Consequently, the network can make use of rich contextual information between the initial predictions in a feature space, which results in less confusion

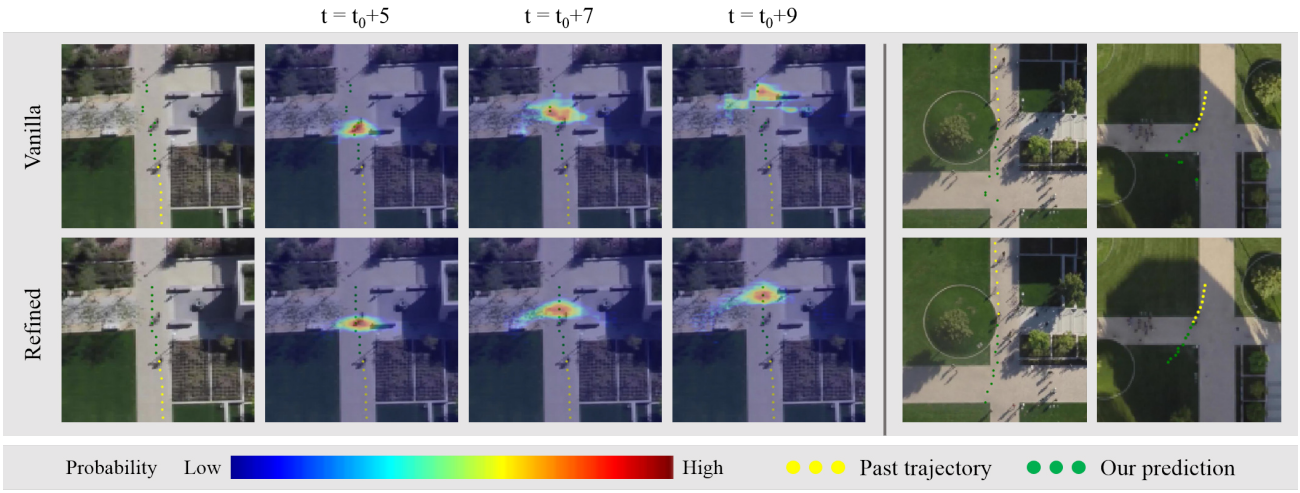
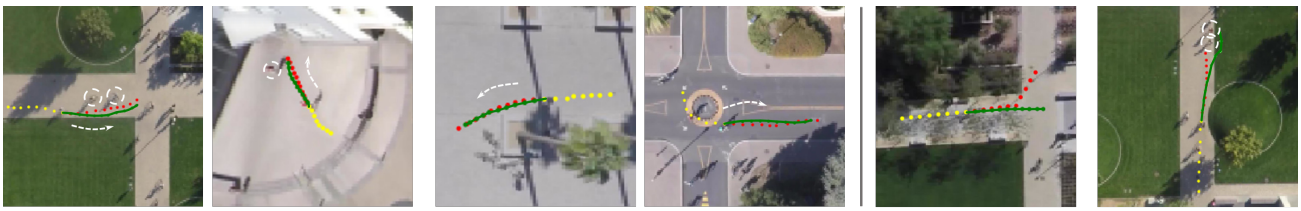


Figure 13: Visual analysis of spatial refinement. The first row shows the predicted future locations from the vanilla trajectory prediction network as presented in Section 4.1. Heatmap predictions are ambiguous, and hence the trajectory is unrealistic. The second row shows the refined locations by considering spatial dependencies as in Section 4.2 in the main manuscript.



(a) (b) (c) (d)

Figure 14: The proposed approach properly encodes (a) human-human and (b) human-space interactions by inferring relational behavior from a physical environment (highlighted by a dashed arrow). However, we sometimes fail to predict a future trajectory when a road user (c) unexpectedly changes the direction of its motion or (d) does not consider the interactions with an environment. (Color codes: Yellow - given past trajectory, Red - ground-truth, and Green - our prediction)

between heatmap locations.

8. Implementation Details

The network models were trained with a GPU (NVIDIA's TITAN Xp) using TensorFlow. We first trained GRE and TPN with stochastic gradient descent and batch size of 30, starting with a learning rate of 5e-4. The learning rate was reduced by a factor of 2 after 100, 150, and 200 epochs. These two networks were trained for 220 epochs, and its performance is reported as *GRE_Vanilla* in the tables in the main muscript. Then, we trained the SRN together with optimizing the rest of the network models using the total loss $\mathcal{L}_{optimize}$ presented in Section 4.3 in the main manuscript. The initial learning rate of 2.5e-5 was reduced by a factor of 2 after 15 epochs, and the network converged after 20 epochs. The performance of the model is evaluated in the tables as *GRE_Refine* and *GRE_MC-L*.

9. Additional Evaluation

9.1. Spatial Refinement

The SRN aims to learn spatial dependencies from initial predictions to enforce the model enforce to generate spatially aligned heatmaps. In addition to Figure 5 in the main manuscript, we further validate its efficacy in Figure 13. With an advantage of the intermediate supervision with two loss functions, the proposed approach shows less confusion for future locations.

9.2. Qualitative Evaluation

Figure 14 further demonstrates that the proposed approach generates natural motion for the target with respect to the consideration of human-human interactions in 14a and human-space interactions in 14b. We also present prediction failures in 14c where the road user suddenly changes course and 14d where the road user is aggressive to interactions with an environment.